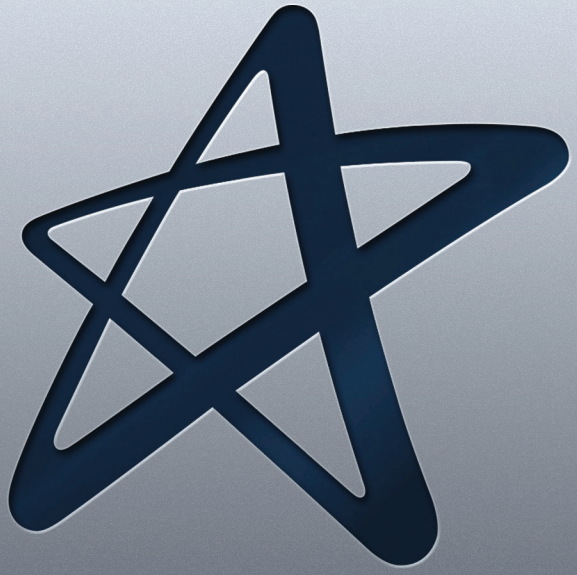
Linguagem de Banco de Dados



Cruzeiro do Sul Virtual
Educação a Distância

Material Teórico

Organização dos dados – Criação de tabelas

Responsável pelo Conteúdo:
Prof. Ms. Luiz Carlos Reis

Revisão Textual:
Prof. Ms. Luciano Vieira Francisco

UNIDADE Organização dos dados – Criação de tabelas

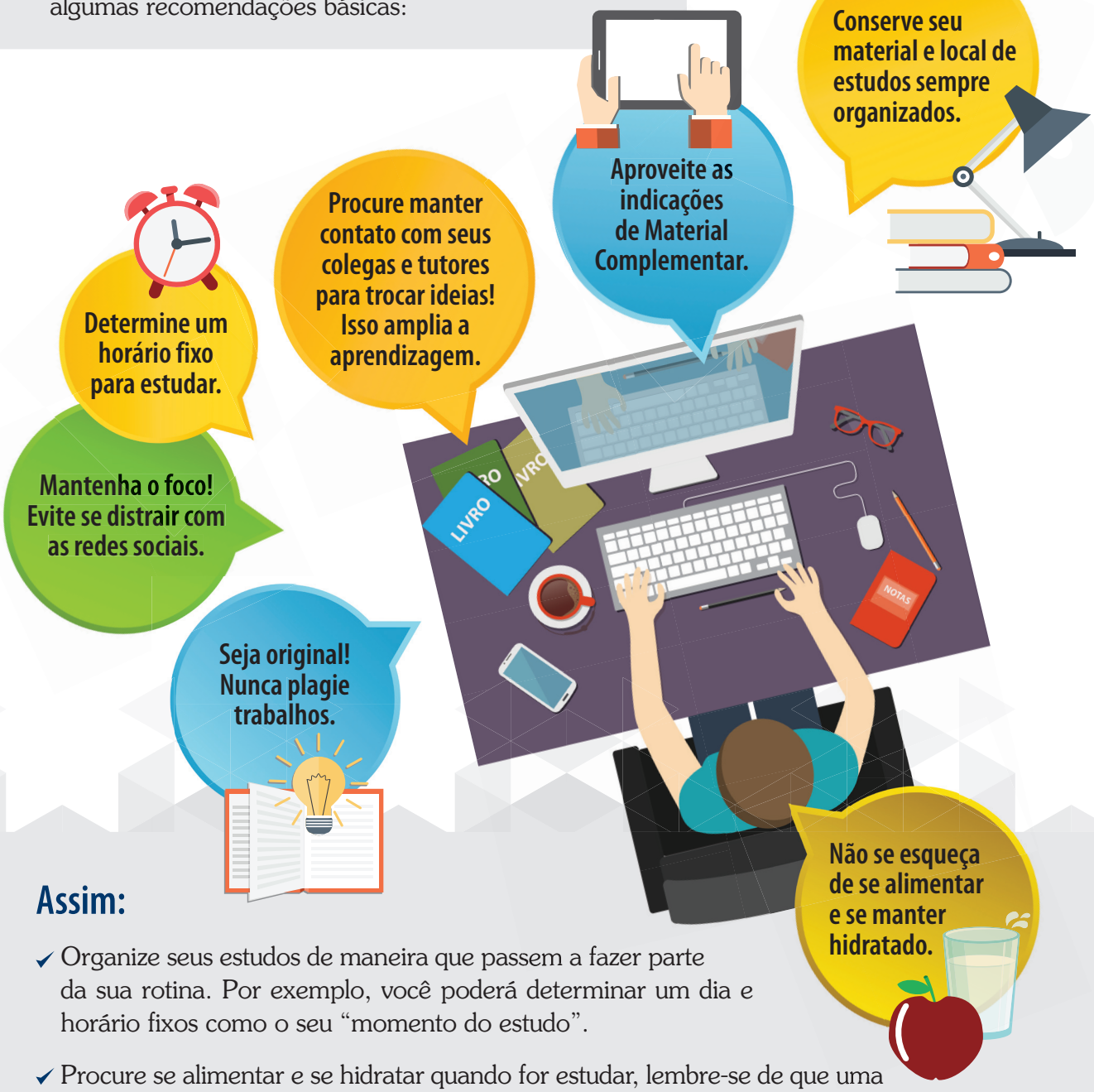
- Introdução
- Tipo de Dados
- Restrições – Constraint
- Restrições – Constraint – de Integridade de Dados
- Criação de Tabela
- Primary Key Composta
- Unique
- Foreign Key
- Not Null
- Alteração de Tabela
- Exclusão de Tabela

OBJETIVO DE APRENDIZADO

- Conhecer o modo de armazenamento interno – tamanho de campos, índices, tipo de preenchimento dos campos, nomenclaturas etc. – através da linguagem denominada SQL.
- Criar a estrutura para armazenar as informações, assim como a definição de regras, tais como chave primária, relacionamentos e restrições.

Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja uma maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:



Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como o seu "momento do estudo".
- ✓ Procure se alimentar e se hidratar quando for estudar, lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo.
- ✓ No material de cada Unidade, há leituras indicadas. Entre elas: artigos científicos, livros, vídeos e *sites* para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados.
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e aprendizagem.

Introdução

Nesta Unidade veremos o comando para a organização de dados, ou seja, o *Data Definition Language* (DDL). Para tanto, utilizaremos o *software* SQLDeveloper como ajuda na prática desses comandos.

Ademais, utilize o tutorial para que você possa configurar o acesso ao banco de dados *Oracle*. A fim de aplicar os seguintes comandos, precisaremos utilizar a conexão *AulaHR*, configurada neste tutorial.

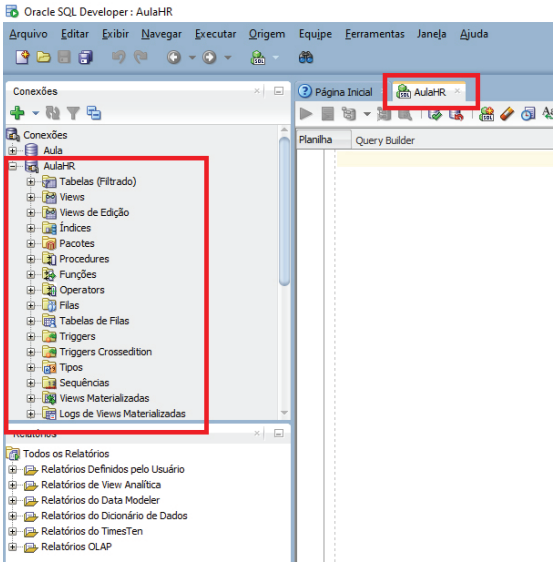


Figura 1

Linguagens e notações variam de acordo com o Sistema de Gerenciamento de Banco de Dados (SGBD), de modo que utilizaremos as regras para o produto da *Oracle*.

SQL significa *Structured Query Language* – linguagem estruturada de consulta – e foi originalmente desenvolvida na *IBM Research*, no início da década de 1970, representando o padrão para linguagens de SGBD relacionais e sendo padronizada pelo comitê Ansi/ISO.

Em 1999, foi publicado o atual padrão SQL/99 ou SQL3. Enquanto os bancos de dados relacionais eram desenvolvidos, foram criadas linguagens destinadas à sua manipulação. O Departamento de Pesquisas da IBM desenvolveu SQL como forma de interface para o sistema de banco de dados relacional, denominado *System R*, no início da década de 1970.

Em 1986, o *American National Standard Institute* (Ansi) publicou um padrão SQL, de modo que este estabeleceu-se como linguagem padrão de banco de dados relacional.

SQL apresenta uma série de comandos que permitem a definição dos dados – a mencionada DDL –, composta, entre outros elementos, pelos comandos *create, alter e drop*.

Importante!

Como utilizaremos o banco de dados da Oracle como auxílio no aprendizado, seguem as regras de nomeação para este tipo de banco de dados:
• Os nomes das tabelas e colunas devem conter de um a trinta caracteres, sendo o primeiro caractere alfabético;
• Os nomes devem conter apenas caracteres de a-z, A-Z, 0-9, _, S e #;
• Não podem ser iguais às palavras reservadas do Oracle

Tipo de Dados

Ao se criar a estrutura de uma tabela, torna-se necessário que o usuário forneça, para cada coluna, as seguintes informações:

- Tipo de dado;
- Tamanho;
- Restrições.

Tabela 1

Tipo de dado	Descrição
CHAR(n)	Campo fixo com tamanho máximo de 2.000 bytes.
DATE	Permite data entre 1 de janeiro de 4712 a.C. até 31 de dezembro de 4712 d.C.
LONG	Caractere variável com tamanho de até 2 GB
VARCHAR2(n)	Campo do tipo caractere com tamanho variável e limitado a 4.000 bytes
NUMBER(n,d)	Onde n é o número de dígitos e d, o número de casas decimais

Fonte: elaborado pelo professor conteudista

Restrições – *Constraint*

As restrições são regras básicas estabelecidas para o preenchimento de uma ou mais colunas da tabela, sendo definida(s) ao final da especificação de cada coluna ou ao final do comando.

Entre as restrições, encontram-se:

- Chaves primárias;
- Chaves únicas;
- Chaves estrangeiras;
- Identificadores de campos obrigatórios;
- Condições para valores permitidos para determinado campo.

Restrições – *Constraint* – de Integridade de Dados

Tabela 2

Constraint	Descrição
NOT NULL	Especifica que esta coluna não pode conter valores nulos
UNIQUE	Especifica uma coluna ou combinação de colunas que terá(ão) seus valores únicos na tabela
PRIMARY KEY	Identifica a unicidade de cada linha na tabela
FOREIGN KEY REFERENCES	Estabelece um relacionamento entre as chaves estrangeira e primária da tabela referenciada
CHECK	Especifica uma condição que deve ser verdadeira, obedecendo a uma regra do negócio

Fonte: elaborado pelo professor conteudista

Criação de Tabela

O comando *CREATE TABLE* cria uma tabela, inicialmente vazia, no banco de dados corrente. Assim, o usuário que executa o comando se torna o dono da tabela. O mesmo comando cria também e automaticamente o tipo de dado que representa a forma composta correspondendo a uma linha da tabela.

Vejamos a sintaxe:

```
CREATE TABLE nome_tabela
( nome_coluna tipo de dado | constraint_tabela ),
( nome_coluna tipo de dado | constraint_tabela )
```

Onde:

Nome_tabela à é o nome da tabela.

Nome_coluna à é o nome da coluna.

Tipo de dado à é o tipo de dado da coluna.

Constraint_tabela à é a "constraint", ou restrição para a coluna.

Tipos de Constraint

Primary Key

Especifica uma ou mais colunas que compõem a chave primária de uma tabela. Uma forma mais organizada, onde após definirmos todos os campos da tabela é definimos as restrições:

```
CREATE TABLE Cliente
(
cd_cliente        number (4),
      nm_cliente        varchar2 (50),
      ds_endereco       varchar2 (70),
```

```
        cd_municipio            number (5),
        sg_estado          char (2),
        nr_cep             varchar2 (8),
        nr_ddd             number (3),
        nr_fone            number(7),
        ie_sexo                 char(1),
              constraint cliente_cd_cliente_pk primary key (cd_
cliente)
)
```

Cole este comando na área de execução e selecione o item destacado em vermelho na seguinte Figura para a sua execução:

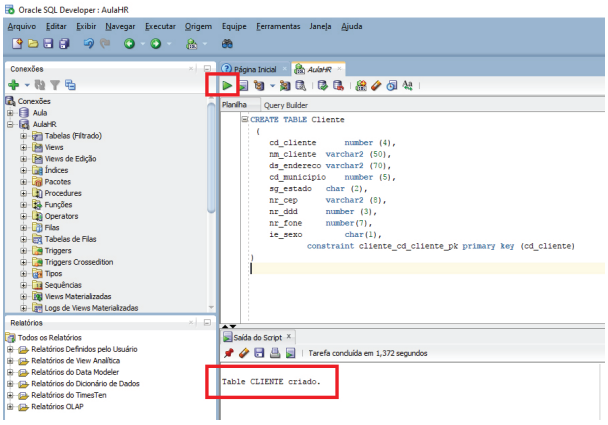


Figura 2

Perceba que a tabela foi criada. Caso não apareça, selecione o item Atualizar.

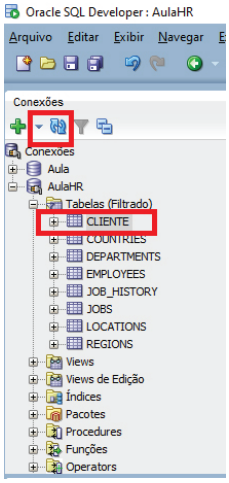


Figura 3

Padronizando o nome da restrição:

Quando criamos uma constraint e não lhe damos um nome, o banco de dados se encarrega de fazê-lo; o problema é que o nome não será nada intuitivo, portanto, seguiremos o seguinte padrão:

```
Constraint=> nome_tabela_nome_campo_tipodaconstraint
```

Importante!

Embora adotemos essa padronização, isso não é regra, de modo que a constraint pode receber qualquer nome.

Primary Key Composta

Uma chave **simples** é associada a um único valor, ou campo da tabela. Já uma chave **composta** corresponde à combinação de dois ou mais campos e pode ser necessária para eliminar a ambiguidade, formando um identificador único.

Um bom exemplo de utilização de chave composta é a junção do campo número da agência e número da conta para uma tabela denominada Conta. Neste exemplo, podemos ter duas contas iguais, mas em agências diferentes e, neste caso, a chave primária é definida pela junção dos campos número da agência e número da conta.

Aqui é apresentado um exemplo de uma *Primary Key* (PK) composta por dois atributos: cd_cliente e dt_compra.

Vejamos:

```
CREATE TABLE  Historico
 (
cd_cliente  number (4),
dt_compra         date,
vl_compra         number (12,2),
CONSTRAINT Historico_PK PRIMARY KEY (cd_cliente, dt_Compra)
)
```

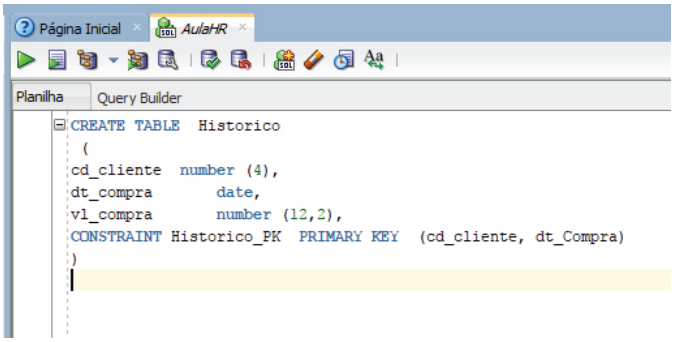


Figura 4

Unique

Você pode usar as restrições *UNIQUE* para ter certeza de que não há valores duplicados digitados em colunas específicas que não participam da chave primária. Embora as restrições *UNIQUE* e PRIMARY KEY imponham exclusividade, use a restrição *UNIQUE* em vez da PRIMARY KEY quando for impor a exclusividade de uma coluna – ou uma combinação de colunas – que não seja uma chave primária.

Várias restrições *UNIQUE* podem ser definidas em uma tabela, ao passo que somente uma restrição PRIMARY KEY pode ser definida em uma tabela. Além disso, ao contrário das restrições PRIMARY KEY, as *UNIQUE* permitem um valor *NULL*. Por exemplo, não existem dois indivíduos com o mesmo número de Cadastro de Pessoas Físicas (CPF) ou número do Programa de Integração Social (PIS), mas tais campos não serão colocados como chaves primárias.

Exemplo:

```
CREATE TABLE Estado
 (
Sg_Estado          char(2) primary key,
      Nm_Estado          varchar2 (35),
      constraint Estado_nm_Estado_UN UNIQUE (nm_Estado)
 )
```

Note que aqui foi garantido que não fossem inseridos, na tabela Estado, os registros duplicados no campo Nome do Estado. Ademais, cole esse comando e execute para criar a tabela.

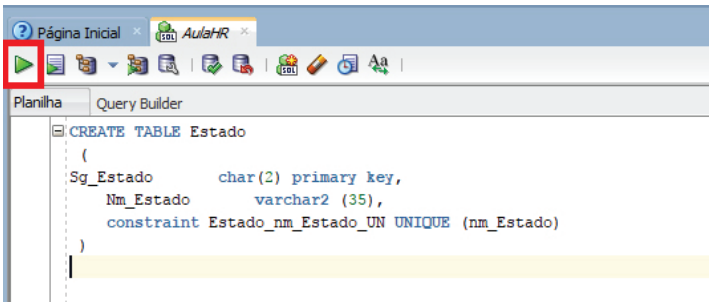


Figura 5

Foreign Key

Chave estrangeira – *foreign key* – é o campo que estabelece o relacionamento entre duas tabelas. Assim, uma coluna corresponde à mesma coluna que é a chave primária de outra tabela. Dessa forma, deve-se especificar na tabela que contém a chave estrangeira quais são essas colunas e à qual tabela está relacionada.

Uma Chave Estrangeira (FK) é uma coluna ou combinação de colunas usadas para estabelecer e impor um *link* entre os dados em duas tabelas. Assim, você pode criar uma chave estrangeira definindo uma restrição *FOREIGN KEY* ao elaborar ou modificar uma tabela.

Em uma referência de chave estrangeira, cria-se um *link* entre duas tabelas quando a(s) coluna(s) que contém(êm) o valor de chave primária para uma tabela é(são) referenciada(s) pela(s) coluna(s) em outra tabela. Essa(s) coluna(s) torna(m)-se chave(s) estrangeira(s) na segunda tabela.

O propósito da restrição *FOREIGN KEY* é controlar os dados que podem ser armazenados na tabela de chave estrangeira, pois os campos nos quais essa faz referência na tabela com chave primária devem estar inseridos, primeiramente.

Para executar novamente a criação da tabela Cliente, exclua a mesma antes de aplicar tal ação.

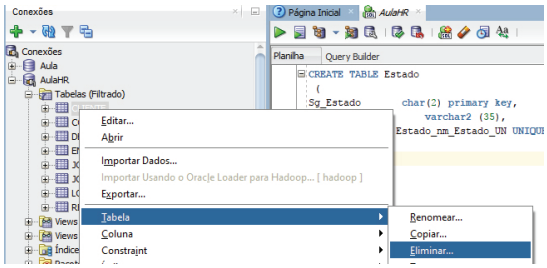


Figura 6

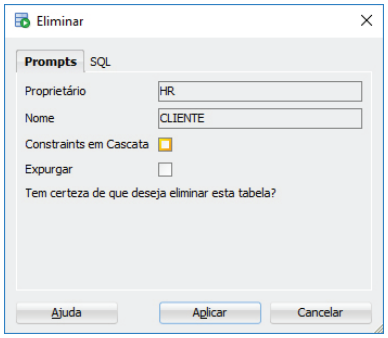


Figura 7

Execute, então e novamente, o seguinte comando para criar a tabela *Cliente* com as regras de FK e PK:

```
CREATE TABLE Cliente
 (
cd_cliente        number (4),
      nm_cliente        varchar2 (50),
      ds_endereco       varchar2 (70),
      cd_municipio            number (5),
      sg_         char (2),
      nr_cep            varchar2 (8),
      nr_ddd            number (3),
      nr_fone           number(7),
      ie_sexo                 char(1),
    constraint cliente_cd_cliente_pk primary key (cd_cliente),
      constraint cliente_sg_estado_fk foreign key (sg_estado)
references Estado(sg_estado)
)
```

Neste exemplo garantimos que não será permitido adicionar uma sigla de estado na tabela Cliente sem, antes, inserir essa sigla na tabela Estado.

Regras:

Caso os tipos de dados da coluna na tabela inicial e na referenciada sejam diferentes, será apresentado um erro. O uso de chaves estrangeiras garante que não existirão linhas órfãs nas tabelas-filhas – que possuem dados que devem estar cadastrados previamente em outra tabela, então denominada tabela-mãe.

Check

Define um conjunto de valores permitidos para um campo ou uma condição a fim de inserir valores em determinado campo. Ademais, antes de criar a tabela, elimine-a novamente, para que assim possamos recriá-la.

Importante!

Caso não apareça a tabela, utilize o botão Atualizar a fim de que seja atualizada a lista das tabelas.

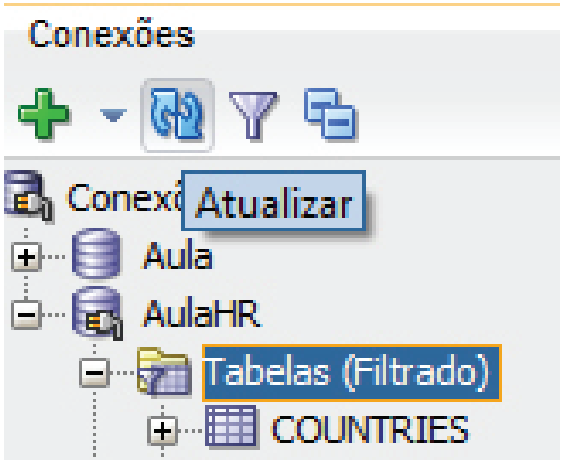


Figura 8

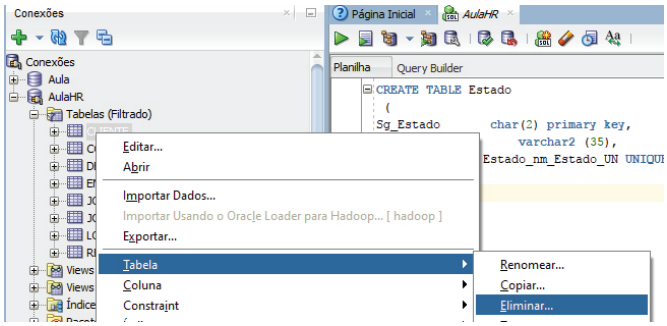


Figura 9

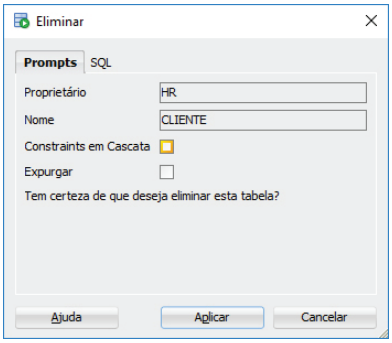


Figura 10

```
CREATE TABLE Cliente
 (
cd_cliente        number (4),
      nm_cliente        varchar2 (50),
      ds_endereco       varchar2 (70),
      cd_municipio            number (5),
      sg_estado         char (2),
      nr_cep            varchar2 (8),
      nr_ddd            number (3),
      nr_fone           number(7),
      ie_sexo                 char(1),
    constraint cliente_cd_cliente_pk primary key (cd_cliente),
      constraint cliente_sg_estado_fk foreign key (sg_estado)
references Estado(sg_estado),
    constraint cliente_ie_sexo_ck check(ie_sexo in ('F', 'M'))
)
```

Com essa regra definida, estará garantido que os registros inseridos apenas poderão ter, no campo sexo, os valores F e M.

Not Null

Indicará a obrigatoriedade para este campo, de modo que caso sejam inseridos registros nessa tabela e não for informado valor para a mesma, o banco de dados retornará um erro. Novamente, antes de executar o seguinte comando, exclua a tabela.

```
CREATE TABLE Cliente
 (
cd_cliente        number (4),
      nm_cliente        varchar2 (50) not null,
      ds_endereco       varchar2 (70) not null,
      cd_municipio            number (5),
      sg_estado         char (2),
      nr_cep            varchar2 (8),
      nr_ddd            number (3),
      nr_fone           number(7),
      ie_sexo                 char(1),
    constraint cliente_cd_cliente_pk primary key (cd_cliente),
      constraint cliente_sg_estado_fk foreign key (sg_estado)
references Estado(sg_estado),
    constraint cliente_ie_sexo_ck check(ie_sexo in ('F', 'M'))
)
```

Perceba a cláusula not *null* na frente do campo nome e endereço, tornando, assim, esses campos obrigatórios.

Alteração de Tabela

Após a criação da estrutura de uma tabela, não é necessário excluí-la para poder modificar a sua estrutura, visto que nessa já podem conter registros. Ademais, podemos querer incluir (ADD), excluir (DROP), modificar (*MODIFY*) colunas ou até constraints dessa tabela, de modo que utilizaremos o comando ALTER para esse propósito. Vejamos alguns exemplos:

- Acrescentar campos em uma tabela:

```
ALTER TABLE CLIENTE
ADD( IE_FISICA_JURIDICA CHAR(1))
```

- Alterando obrigatoriedade de atributos em uma tabela:

```
ALTER TABLE CLIENTE
MODIFY (nm_cliente not null);
```

- Modificando o tipo de atributos em uma tabela:

```
ALTER TABLE CLIENTE
MODIFY (DS_ENDERECO NUMBER(3));
```

- Modificando o tamanho dos atributos em uma tabela:

```
ALTER TABLE CLIENTE
MODIFY (NR_CEP VARCHAR2(12));
```

- Acrescentar restrições a uma tabela:

```
ALTER TABLE CLIENTE
ADD(CONSTRAINT CLIENTE_IE_FISICA_JURIDICA_CK
CHECK(IE_FISICA_JURIDICA IN('F','J')));
```

- Desabilitar uma restrição de uma tabela:

```
ALTER TABLE CLIENTE
DISABLE CONSTRAINT  CLIENTE_IE_FISICA_JURIDICA_CK;
```

- Excluir uma restrição de uma tabela:

```
ALTER TABLE CLIENTE
DROP CONSTRAINT CLIENTE_IE_FISICA_JURIDICA_CK
```

Exclusão de Tabela

Após a criação ou alteração da estrutura de uma tabela, podemos excluir a mesma por meio do comando DROP.

Ao excluir a tabela, todas as constraints e os dados inseridos serão fisicamente deletados.

Vejamos:

```
DROP TABLE CLIENTE.
```

Material Complementar

Indicações para saber mais sobre os assuntos abordados nesta Unidade:

Livros

Oracle Database 11g Sq

Como complemento a esta Unidade, leia o terceiro capítulo do livro de J. Price, intitulado ***Oracle* Database 11g Sql**, publicado em 2009, pela editora Bookman, de Porto Alegre, RS.

SQL : guia prático

Ler do capítulo II do livro: Costa, Rogério Luis de C. **SQL** : guia prático. 2. ed. Rio de Janeiro : Brasport, 2006.

Vídeos

Criação de Tabelas

https://youtu.be/rIlzcwlCts0

Aula 2 - Criando as tabelas - ORACLE

https://youtu.be/lg9s_Y5GSig

Referências

COSTA, R. L. de C. **SQL**: guia prático. 2. ed. Rio de Janeiro: Brasport, 2006.

MORELLI, E. M. T. ***Oracle* 9i fundamental**: Sql, Pl/SQL e administração. São Paulo: Érica, 2002.

PRICE, J. ***Oracle* Database 11g Sql**. Porto Alegre, RS: Bookman, 2009.

SILBERSCHATZ, A. **Sistema de bancos de dados**. São Paulo: Pearson Education do Brasil, 2004.

Cruzeiro do Sul Virtual
Educação a Distância

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo - SP - Brasil
Tel: (55 11) 3385-3000

Cruzeiro do Sul
Educacional